

# Simulation of many-qubit quantum computation with matrix product states

M. C. Bañuls,<sup>1</sup> R. Orús,<sup>2</sup> J. I. Latorre,<sup>2</sup> A. Pérez,<sup>1</sup> and P. Ruiz-Femenía<sup>3</sup>

<sup>1</sup>*Dept. Física Teòrica and IFIC, U. València - CSIC, 46100 Burjassot, València, Spain.*

<sup>2</sup>*Dept. d'Estructura i Constituents de la Matèria, Univ. Barcelona, 08028, Barcelona, Spain.*

<sup>3</sup>*Max-Planck-Institut für Physik (Werner-Heisenberg-Institut), Föhringer Ring 6, 80805 München, Germany.*

Matrix product states provide a natural entanglement basis to represent a quantum register and operate quantum gates on it. This scheme can be materialized to simulate a quantum adiabatic algorithm solving hard instances of a NP-Complete problem. Errors inherent to truncations of the exact action of interacting gates are controlled by the size of the matrices in the representation. The property of finding the right solution for an instance and the expected value of the energy (cost function) are found to be remarkably robust against these errors. As a symbolic example, we simulate the algorithm solving a 100-qubit hard instance, that is, finding the correct product state out of  $\sim 10^{30}$  possibilities. Accumulated statistics for up to 60 qubits seem to point at a sub-exponential growth of the average minimum time to solve hard instances with highly-truncated simulations of adiabatic quantum evolution.

PACS numbers: 03.67.-a, 03.65.Ud, 03.67.Hk

A detailed understanding of a many-spin quantum system often requires its simulation on a classical computer. Such a possibility is limited to a small number of spins due to the exponential growth of the size of the Hilbert space. This is at the heart of the motivation to build a quantum computer [1]. Using standard present technology, a faithful simulation of a general Hamiltonian can be achieved for systems up to the order of 24 spins.

Recent developments in representing quantum states and operating unitary evolution on them have refined the above common lore. The idea has evolved from accumulated knowledge on matrix product states (MPS, related to the density matrix renormalization group technique) [2] and new insights from quantum information. Let us recall that a quantum state for an  $n$ -qubit system can be represented by the matrix product construction

$$|\psi\rangle = \sum_{\{i\}} \sum_{\{\alpha\}} A_{1\alpha_1}^{(1)i_1} A_{\alpha_1\alpha_2}^{(2)i_2} \dots A_{\alpha_{n-1}\alpha_n}^{(n)i_n} |i_1, i_2, \dots, i_n\rangle, \quad (1)$$

where the indices  $i_1, \dots, i_n$  for each qubit range from 0 to 1 (the qubits are placed in a chain) and  $\alpha_1, \dots, \alpha_{n-1}$  are referred to as ancillae indices that range from 1 to a parameter we shall call  $\chi$ . Each matrix  $A_{\alpha_{a-1}\alpha_a}^{(a)i_a}$  at site  $a$  can be viewed as a projector from a pair of unphysical ancillae to the physical degree of freedom that we associate to the computational basis. The success of MPS consists in changing the representation of the quantum state from the computational basis to a non-local one, closely attached to entanglement. To make this comment concrete, let us note that the matrix representation of a state can be recovered via a chain of Schmidt decompositions that separate a local system at a time, as made explicit by Vidal [3]. More specifically,  $A_{\alpha_{a-1}\alpha_a}^{(a)i_a} = \Gamma_{\alpha_{a-1}\alpha_a}^{(a)i_a} \lambda_{\alpha_a}^{(a)}$ ,  $\lambda_{\alpha_a}^{(a)}$  being the Schmidt coefficients of the cut of the system between the  $a$  and  $a+1$  sides, and  $\Gamma^{(a)}$  being tensors for qubit  $a$ . The larger the entanglement is for different

partitions of the system, the larger is the needed ancillae space, which corresponds to a higher rank  $\chi$ . MPS can handle simulations of various dynamics of spin chains with up to hundreds of spins because their little amount of entanglement can be represented with  $\chi = O(\text{poly}(n))$  [3, 4]. A number of new developments have popped up from the basic MPS in the context of quantum information. In ref. [3], an efficient implementation of Hamiltonian evolution was constructed for slightly entangled systems. An explicit renormalization group transformation on quantum states was made explicit using MPS [5]. The rigid linear structure of MPS is being now abandoned in favor of the more general projected entangled-pair states (PEPS) that have been successfully applied to higher dimensional systems [6].

The natural question arises of whether MPS can be applied to simulate a quantum computer. The content of this paper is aimed to show that this is indeed possible and that we can handle large simulations with controlled accuracy. As we shall describe, each time an entangling gate is operated on two neighboring qubits, the range of the connected ancillae index is doubled. This is the way interacting gates entangle the system. To keep the simulation under control, a (non-unique) truncation scheme is needed that stops the exponential growth of ancillae dimensions. We expect this approximation scheme to fail whenever the inherently needed  $\chi$  is  $O(2^n)$ . Nevertheless, in some of these cases keeping  $\chi = O(\text{poly}(n))$  in the simulation already gives reasonable approximations to the exact calculation, as we shall see.

Our presentation will be made concrete by showing an MPS simulation of quantum computation in the case of adiabatic evolution for the NP-Complete Exact Cover satisfiability problem [7, 8]. An instance of Exact Cover is defined by a set of  $m$  3-bit clauses with satisfying assignments 001, 010 or 100. The problem is defined as

deciding whether a given instance accepts a global satisfying assignment of  $n$  bits. This satisfiability problem is NP-Complete. Classically hard instances of Exact Cover seem to appear at the so-called easy-hard-easy transition around  $m \sim .8n$  [9]. We have constructed such hard instances, with the additional property of having a unique satisfying assignment. The generation of hard instances is in itself a difficult problem for which we have developed specific algorithms, all of them based on the iterative addition of random clauses that strictly decrease the number of solutions of the instance until a single satisfying assignment is reached.

The quantum algorithm for a given Exact Cover instance follows the adiabatic evolution of the ground state of a Hamiltonian (cost operator) defined by  $H(s) = (1-s)H_0 + sH_P$ , where the adiabatic parameter is  $s = t/T$  and  $t$  runs up to a total predetermined time  $T$ . We take the initial Hamiltonian to be  $H_0 = \sum_{i=1}^n \frac{d_i}{2}(1-\sigma_i^x)$  where  $d_i$  stands for the number of clauses qubit  $i$  enters. The non-local problem Hamiltonian corresponds to the sum of clauses defined as  $H_P = \sum_{c(i,j,k)} (z_i + z_j + z_k - 1)^2$  where  $z_i = (1 - \sigma_i^z)/2$  has eigenvalues 0 and 1, and  $c(i, j, k)$  stands for a clause involving qubits  $i, j$  and  $k$ . Exact simulations of quantum algorithms by adiabatic evolution solving hard instances of satisfiability problems have been carried so far up to 30 qubits [10]. The explosion of entanglement between random cuts in the quantum register was first analyzed in ref. [11]. The adiabatic evolution drives the system near a quantum phase transition at  $s \sim .69$  following universal scaling laws. Entropy for half-cuts of the register approximates on average the scaling law  $S \sim .1n$ , which almost saturates the maximum  $S = n/2$ . This implies that the quantum algorithm cannot be simulated efficiently in a classical computer [3]. Yet, the fact that entropy does not reach its allowed maximum suggests that an adequate handling of entanglement may provide a way to extend simulations far from naive limitations.

Let us now turn to discuss the detailed way MPS can handle the simulation of the adiabatic evolution of Exact Cover. The simulation needs to follow a time evolution controlled by the  $s$ -dependent Hamiltonian. This continuous unitary time evolution can be discretized as follows:  $U_{T,0} = U_{T,T-\Delta} \dots U_{2\Delta,\Delta} U_{\Delta,0}$  where the increment  $\Delta \equiv \frac{T}{M}$  defines the discretization,  $M$  being a positive integer. Our simulations indicate that we can take  $\Delta = 0.125$  while keeping sufficient accuracy (as compared to smaller  $\Delta$ ) in all of them. We have explicitly checked that simulations performed with  $\Delta < 0.125$  lead to equally-good discretizations of the continuous-time adiabatic algorithm, in the sense that the obtained results do not practically differ from the ones calculated for  $\Delta = 0.125$ . After  $l$  steps  $s = \frac{t}{T} = \frac{l\Delta}{T} = \frac{l}{M}$ , being  $l = 0, \dots, M$ . At any point in the evolution Trotter's formula to second order is used to divide the unitary op-

eration  $U_{(l+1)\Delta,l\Delta}$  into elementary gates:  $U_{(l+1)\Delta,l\Delta} = e^{i\Delta H(s)} \sim \left( e^{i\frac{\Delta}{2}(1-s)H_0} e^{i\Delta s H_P} e^{i\frac{\Delta}{2}(1-s)H_0} \right)^{\frac{1}{\Delta}}$ . We have verified that we can maintain a faithful simulation with  $\delta = \Delta$ . The split of exponentials in Trotter's expansion is chosen so that  $H_0$  is separated from  $H_P$ . This brings the advantage that each piece of the evolution operator can be decomposed in mutually commuting elementary gates:

$$e^{i\frac{\Delta}{2}(1-s)H_0} = \prod_{i=1}^n e^{i\frac{\Delta}{4}(1-s)d_i(1-\sigma_i^x)}, \quad (2)$$

and

$$\begin{aligned} e^{i\Delta s H_P} &= \prod_{c(i,j,k)} e^{i\Delta s (z_i + z_j + z_k - 1)^2} \\ &= \prod_{c(i,j,k)} e^{i\Delta s (z_i^2 - 2z_i)} e^{i\Delta s (z_j^2 - 2z_j)} e^{i\Delta s (z_k^2 - 2z_k)} e^{i\Delta s} \\ &\quad e^{i2\Delta s z_i z_j} e^{i2\Delta s z_i z_k} e^{i2\Delta s z_j z_k}. \end{aligned} \quad (3)$$

The adiabatic evolution is thus finally reduced to a series of one and two-qubit gates. The detailed way these gates operate on the MPS follows the original idea of ref. [3]:

1. A one-qubit gate acting on qubit  $a$  only involves an updating of  $A^{(a)}$  that goes as follows:

$$U^{(a)} A_{\alpha\beta}^{(a)i_a} |i_a\rangle = A_{\alpha\beta}^{(a)i_a} U_{i_a i'_a}^{(a)} |i'_a\rangle, \quad (4)$$

which corresponds to the local updating rule

$$A_{\alpha\beta}'^{(a)i'_a} = U_{i_a i'_a}^{(a)} A_{\alpha\beta}^{(a)i_a}. \quad (5)$$

This gate does not affect ancillae indices. Entanglement is unaffected as we are just performing local operations.

As an example, consider the one-qubit gate  $U^{(a)} = \sigma_x^{(a)}$ ,  $\sigma_x^{(a)}$  being the usual Pauli matrix

$$\sigma_x^{(a)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (6)$$

acting on qubit  $a$ . Then, we have the following simple updating rule for  $A^{(a)}$ :

$$\begin{pmatrix} A_{\alpha\beta}'^{(a)0} \\ A_{\alpha\beta}'^{(a)1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} A_{\alpha\beta}^{(a)0} \\ A_{\alpha\beta}^{(a)1} \end{pmatrix} = \begin{pmatrix} A_{\alpha\beta}^{(a)1} \\ A_{\alpha\beta}^{(a)0} \end{pmatrix}. \quad (7)$$

2. A two-qubit gate involving contiguous qubits  $a$  and  $a+1$  follows a similar strategy. Let us define

$$U_{i'_a i'_{a+1}, i_a i_{a+1}}^{(a,a+1)} A_{\alpha\beta}^{(a)i_a} A_{\beta\gamma}^{(a+1)i_{a+1}} \equiv \Theta_{\alpha\gamma}^{i'_a i'_{a+1}}. \quad (8)$$

Unlike one-qubit gates, interacting gates do not preserve the product form of the tensors  $A$ . To reestablish the MPS structure we need to rewrite  $\Theta$  using a Schmidt decomposition. The procedure to follow is to compute the reduced density matrix from the cut of the system

between the  $a$  and  $a + 1$  sites, which for the right side reads  $\rho_{\alpha\gamma}^{ij} = |\lambda_{\beta}^{(a-1)}|^2 \Theta_{\beta\alpha}^{ki} \Theta_{\beta\gamma}^{*kj}$ , where we have made use of the  $\chi$  known Schmidt coefficients  $\lambda_{\beta}^{(a-1)}$  for the cut between the  $a - 1$  and the  $a$  sites. After the diagonalization of  $\rho$  using  $(i\alpha)$  and  $(j\gamma)$  as composed indices, we directly read from the eigenvalues the updated  $2\chi$  Schmidt coefficients  $\lambda'_{\beta}^{(a)}$  for this cut, and the updated matrices  $A'_{\beta\gamma}^{(a+1)i_{a+1}}$  from the coefficients of the eigenvectors. Finally, the new tensors for qubit  $a$  are easily calculated as  $A'_{\alpha\beta}^{(a)i_a} = A'_{\beta\gamma}^{(a+1)i_{a+1}} \Theta_{\alpha\gamma}^{i_a i_{a+1}}$ .

Let us clarify this procedure with a simple example: consider the quantum state of two qubits

$$|\psi\rangle = |00\rangle. \quad (9)$$

It is easy to verify that the above state is described by the following values of the matrices  $A^{(a)}$ :

$$\begin{aligned} A_{1\alpha}^{(1)0} &= A_{\alpha 1}^{(2)0} = \delta_{1,\alpha} \\ A_{1\alpha}^{(1)1} &= A_{\alpha 1}^{(2)1} = 0. \end{aligned} \quad (10)$$

Notice that since the state is separable  $\chi = 1$ . At this point, let us apply the two-qubit gate

$$U^{(1,2)} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad (11)$$

to the quantum state  $|\psi\rangle$ :

$$U^{(1,2)}|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \quad (12)$$

Since the resultant state is a maximally entangled state of 2 qubits, we expect  $\chi$  to be bigger than 1. In order to evaluate the updated matrices  $A'^{(a)}$  for qubits 1 and 2 we compute the quantity defined in equation (8), which in our case turns out to be

$$\begin{aligned} \Theta^{00} &= \Theta^{11} = \frac{1}{\sqrt{2}} \\ \Theta^{01} &= \Theta^{10} = \frac{1}{\sqrt{2}}. \end{aligned} \quad (13)$$

The density matrix for qubit 2 (which in this case is equivalent to the density matrix for qubit 1) then reads

$$\rho = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}. \quad (14)$$

Since the above density matrix is already diagonal, it is clear that the updated Schmidt coefficients will be

$$\lambda'_1{}^{(1)} = \lambda'_2{}^{(1)} = \frac{1}{\sqrt{2}} \quad (15)$$

and, as expected, we see that  $\chi = 2$  since entanglement has been created by the two-qubit gate. From the above

expressions it is simple to get the value of the updated matrices  $A'^{(a)}$  for qubits 1 and 2:

$$\begin{aligned} A'_{11}{}^{(1)0} &= A'_{11}{}^{(2)0} = 1 \\ A'_{12}{}^{(1)1} &= A'_{21}{}^{(2)1} = 1 \\ A'_{12}{}^{(1)0} &= A'_{21}{}^{(2)0} = 0 \\ A'_{11}{}^{(1)1} &= A'_{11}{}^{(2)1} = 0. \end{aligned} \quad (16)$$

3. Operations involving non-contiguous qubits (as in Exact Cover clauses) can be reduced to the case 2 using SWAP operations, producing an overhead of  $O(n)$  operations per clause.

The exact simulation of a quantum computer is then completely defined. The running time of this algorithm scales as  $\sim Tnm\chi^3$ . Efficiency depends on the way the growth of the ancillae space is handled. To keep the simulation under control we define a truncation scheme of the exact simulation. We choose to use a local procedure, namely, we keep the first  $\chi$  terms out of the  $2\chi$  in the Schmidt decomposition defined in the point 2 above. Only the terms that carry most of the entanglement in the decomposition are kept [3]. This reasonable truncation carries an inherent -but always under control- loss of unitarity, since the sum of the retained squared eigenvalues will not reach 1. As we shall see, larger  $\chi$ 's allow for more faithful simulations. Alternatively, it would be possible to recast the whole enlarged state into its original size in an optimal way [6]. While this second method is manifestly more precise, it carries an operational time overhead. It is then worth analyzing both techniques. In this paper we shall focus on the first one and leave the second for a separate publication.

We have implemented a number of optimizations upon the above basic scheme. For any non-local gate there is an overhead of SWAP operations that damage the precision of the computation. To minimize this effect, every three-qubit clause is operated as follows: we bring together the three qubits with SWAPs of the left and right qubits keeping the central one fixed and, then, we operate the two-qubit gates. Before returning the qubits to their original position we check if any of them is needed in the next gate. If so, we save whatever SWAP may be compensated between the two gates. Ordering of gates is also used to produce a saving of  $\sim 2/3$  of the naive SWAPs. Diagonalization of the density matrix in the minimum allowed Hilbert space is used as well. A further improvement is to keep a dynamical and local  $\chi$ , so that ancillae indices at the different partitions are allowed to take independent values and grow up to site-dependent limits. This procedure, though, has shown essentially no improvement upon a fixed- $\chi$  strategy.

Let us now focus on the results. We first simulate the adiabatic algorithm with the requirement that the

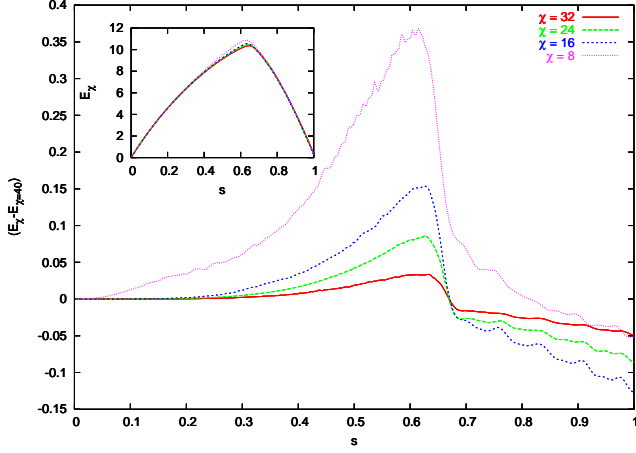


Figure 1: (Color online) Computation of the absolute error (compared to the  $\chi = 40$  case) of the expected value of the energy (cost function) along the adiabatic evolution for a typical instance with 30 qubits and 24 clauses for  $T = 100$  as  $\chi$  increases. Note the increasing precision with larger  $\chi$  as  $s$  approaches the phase transition from the left-hand-side. In the inset, the absolute cost function is plotted. A similar behavior is also obtained for other instances, all of them getting the exact solution at the end of the computation.

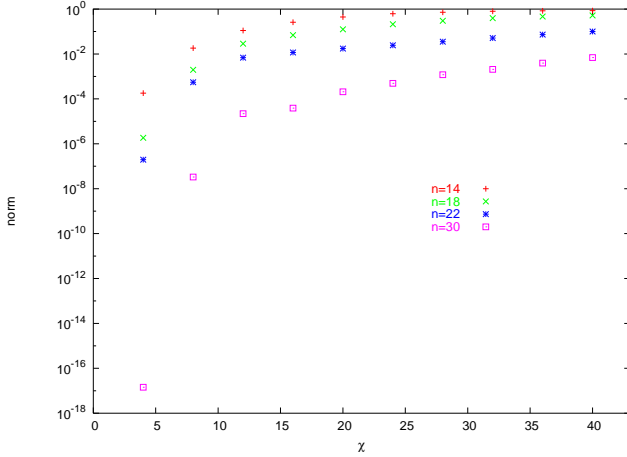


Figure 2: (Color online) Norm in the register as a function of  $\chi$  in logarithmic scale, for instances of 14, 18, 22 and 30 qubits.

right solution is found for a typical instance of  $n = 30$  qubits with  $m = 24$  clauses and  $T = 100$ . Along the evolution we compute the expected value of the energy (cost function) of the system, which can be calculated in  $O(n \text{ poly}(\chi))$  time. This is shown in Fig. 1. The system remains remarkably close to the instantaneous ground-state cost function along the approximated evolution. The error in the cost function is minimized as  $\chi$  increases. It is noteworthy to observe how the error in the simulation of the adiabatic algorithm peaks at the phase transition point. We have also checked that it is precisely at this point where each qubit makes a decision towards

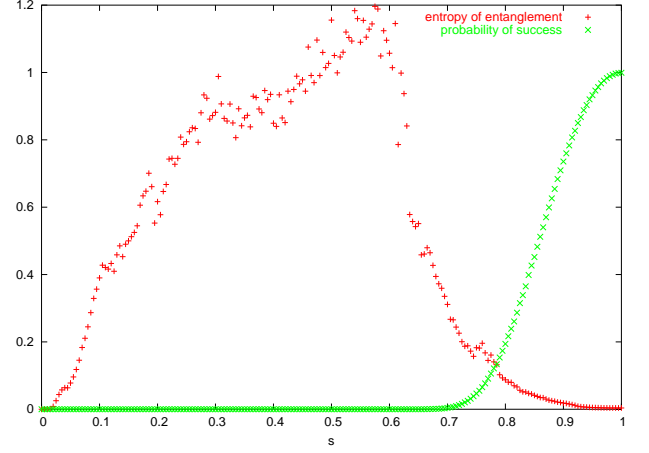


Figure 3: (Color online) An instance with  $n = 100$  qubits and  $m = 84$  clauses is solved using adiabatic evolution with  $\chi = 14$ . The plot shows the entanglement entropy of a half-cut and the probability in solution state vs.  $s$ .

its final value in the solution. Physically, the algorithm builds entanglement up to the critical point where the solution is singled out and, thereon, the evolution drops the superposition of wrong states in the register.

This success comes at the price of a controlled loss of unitarity. We plot in Fig. 2 the norm in the simulation as a function of  $\chi$  in logarithmic scale, for instances of 14, 18, 22 and 30 qubits. The remarkable fact is that some observables, like the energy, appear to be very robust against this inaccuracy. Our simulations also allow to compute the decay of the  $\chi$  Schmidt coefficients  $\lambda_\alpha^{(a)}$  at any step of the computation. Close to criticality, and for the central cut of the system, these can be approximately fitted by the law  $\log_2(\lambda_\alpha^{(n/2)}) = b + \frac{c}{\sqrt{\alpha}} + d\sqrt{\alpha}$ , with appropriate coefficients  $b, c$  and  $d$ .

The ultimate goal of finding the correct solution appears also to be very robust in the simulations we have performed. The exact probability of success can be calculated in  $O(n \text{ poly}(\chi))$  time as well. As a symbolic example, our program has solved an instance with  $n = 100$  qubits, that is, the adiabatic evolution algorithm has found the correct product state out of  $2^{100} \sim 10^{30}$  for a hard instance with  $m = 84$  clauses and  $T = 2000$ . The simulation was done with a remarkable small  $\chi = 14 \ll 2^{50} = \chi_{max}$  and is presented in Fig. 3.

The same robustness of evolving towards the correct solution is found for any number of qubits and small  $\chi$ . We have launched a search for the minimal  $T$  that solves samples of  $n$ -qubit hard instances in the following way: for a set of small values of  $\chi$ , we try a random instance with an initial e.g.  $T = 100$ . If the solution is found, we proceed to a new instance, and if not, we restart with a slower adiabatic evolution e.g.  $T = 200$ . This slowing down of the algorithm is performed till a correct solu-

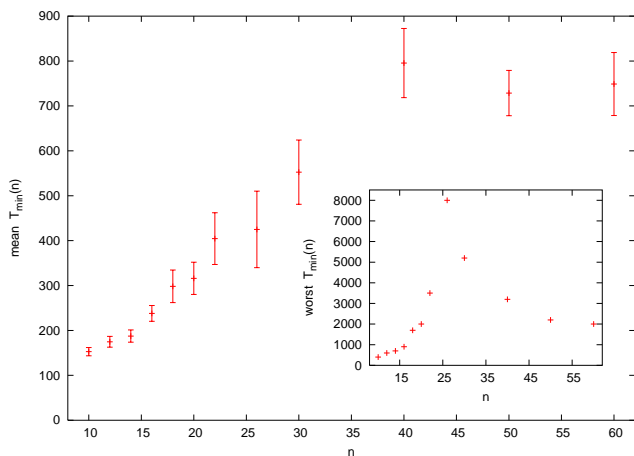


Figure 4: (Color online) Accumulated statistics up to  $n = 60$  for  $T_{min}(n)$  for which each instance is solved (mean and worst cases). Averages are performed over 200 instances for each  $n$ , except for  $n = 50, 60$  with respectively 199, 117 instances. Error bars give 95 per cent of confidence level in the mean. The worst cases found are shown in the inset.

tion is found and the minimal successful  $T_{min}$  is stored. Our results are shown in Fig. 4. The average over  $n$ -qubit instances appears to grow sub-exponentially with  $n$ . In fact, a quadratic fit reproduces the data for  $n < 22$ , consistently with the results found in [7]. The required times for larger  $n$  lie below the extrapolated curve. Isolated instances, however, may require larger times. Since the worst  $T_{min}$  found depends on the interpolating path, finding an instance that needs a very large  $T_{min}$  is no counterproof for the efficiency of the adiabatic algorithm, as alternative paths may solve the instance in a shorter  $T$  [7].

In this paper we have presented simulations of quantum computation based on matrix product states that can be taken up to 100 qubits. The remarkable fact that the algorithm finds the correct solution to a large hard instance and the robustness in the expected energy is to be contrasted with the loss of unitarity inherent to

the local truncation scheme. This drawback may well be ameliorated if optimal truncations are implemented.

**Acknowledgments:** we acknowledge discussions with I. Cirac, E. Farhi and G. Vidal, and support from FPA2001-3598, GC2001SGR-00065, GVA2005-264 and FPA2002-00612. We would like to express our gratitude for the use of the GRID computing resources (GoG farm) and the support of computing technical staff of IFIC.

- 
- [1] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
  - [2] A. Affleck, T. Kennedy, E.H. Lieb, H. Tasaki, Commun. Math. Phys. **115**, 477 (1988); S. R. White, Phys. Rev. Lett. **69**, 2863 (1992); M. Fannes, B. Nachtergaele, R. F. Werner, Commun. Math. Phys. **144**, 443 (1992); S. Ostlund, S. Rommer, Phys. Rev. Lett. **75**, 3537 (1995); J. Dukelsky, M. A. Martín-Delgado, T. Nishino, G. Sierra, Europhys. Lett. **43**, 457 (1998); F. Verstraete, D. Porras, J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004).
  - [3] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003); G. Vidal, Phys. Rev. Lett. **93**, 040502 (2004).
  - [4] G. Vidal, J. I. Latorre, E. Rico, A. Kitaev, Phys. Rev. Lett. **90**, 227902 (2003); J. I. Latorre, E. Rico, G. Vidal, Quant. Inf. and Comp. **4**, 1 (2004).
  - [5] F. Verstraete, J.I. Cirac, J.I. Latorre, E. Rico, M.M. Wolf, Phys. Rev. Lett. **94** 140601 (2005).
  - [6] F. Verstraete, J.I. Cirac, cond-mat/0407066.
  - [7] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, quant-ph/0001106; E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, quant-ph/0104129; E. Farhi, J. Goldstone, S. Gutmann, quant-ph/0208135.
  - [8] M. R. Garey, D. S. Johnson, *Computers and intractability; a guide to the Theory of NP-Completeness*, W. H. Freeman and company, 1979.
  - [9] A clause of Exact Cover is built from 5 clauses of 3-SAT, for which the transition is around  $m \sim 4.26n$ . See J. M. Crawford, L. D. Auton, Artificial Intelligence **81**:31-57 (1996).
  - [10] T. Hogg, Phys Rev A **67** 022314 (2003).
  - [11] R. Orús, J. I. Latorre, Phys. Rev. A **69**, 052308 (2004); J. I. Latorre, R. Orús, Phys. Rev. A **69**, 062302 (2004).